

Vizualizace interiéru pomocí dekorací rozšířené reality

Augmented Reality Application for Interior Design

Miroslava Mačejková

Bakalářská práce

Vedoucí práce: Mgr. Ing. Michal Krumník, Ph.D.

Ostrava, 2021

Abstrakt

Témou tejto práce je rozšírená realita pre interiérový dizajn. Aplikácia bola vyvíjaná pre Android. V úvodných kapitolách sa zoznámite s históriou a vznikom. Ďalej popisuje rozdiel medzi rozšírenou a virtuálnou realitou ako aj nástroje, ktoré je možné zvoliť pre vývoj rozšírenej reality. Následne práca popisuje praktickú implementáciu s využitím programovacieho jazyka java, softvérového rámca AR-Core a vývojárskeho nástroja Sceneform. V ďalších kapitolách je uvedený postup pri tvorbe aplikácie s názvom „Vizualizácia interiéru pomocou dekorácií rozšírenej reality“. Záverečná kapitola obsahuje testovanie aplikácie.

Klíčové slová

mobilná aplikácia, Android, rozšírená realita

Abstract

The topic of this thesis is augmented reality application for interior design. The application was developed for Android. In the first chapters, the thesis describes of history and development augmented reality. Described further, the difference between augmented reality and virtual reality. Subsequently, the work describes the practical implementation using the java programming language, the ARCore software framework and the Sceneform development tool. The next chapters show how to create an application called „Augmented Reality Application for Interior Design“. In the end of the thesis, the final chapter contains application testing.

Keywords

mobile application, Android, augmented reality

Pod'akovanie

Rada by som sa poďakovala Mgr. Ing. Michalovi Krumniklovi, PhD. za odbornú pomoc a konzultáciu pri tvorbe tejto bakalárskej práce.

Obsah

Zoznam použitých symbolov a skratiek	6
Zoznam obrázkov	7
Zoznam tabuliek	8
1 Úvod	9
2 Analýza Frameworkov	10
2.1 Rozdiel medzi frameworkom a SDK	10
2.2 Rozšírená realita/Predstavenie virtuálnej reality	10
2.3 História a vývoj	11
2.4 AR Core a Sceneform SDK	19
3 Analýza a návrh aplikácie	21
3.1 Požiadavky na aplikáciu	21
3.2 Role	22
3.3 Use-Case diagram	22
3.4 Voľba nástroja pre vývoj	22
3.5 Prvotný návrh UI	23
4 Implementácia aplikácie	25
4.1 Použité knižnice	25
4.2 Technické požiadavky	25
4.3 Vývoj aplikácie	26
4.4 Pripojenie	26
4.5 Renderovanie 3D obrázku	27
5 Testovanie	29
5.1 Nasadenie aplikácie	30
5.2 Testovanie pri vývoji	30

5.3	Priebeh testovania na platforme Android	31
5.4	Užívateľské testovanie	31
5.5	Konečné hodnotenie testovania	32
6	Záver	33
	Literatúra	35
	Prílohy	36

Zoznam použitých skratiek a symbolov

Java	– Objektovo orientovaný programovací jazyk
ARCore	– Framework pre rozšírenú realitu
API	– Rozhranie pre programovacie jazyky
APK	– Balík android
Command line	– Príkazový riadok
CPU	– Centrálna procesová jednotka
Framework	– Softvérový rámec
GPU	– Grafický procesor
GPS	– Globálny lokalizačný systém
Headset	– Slúchadlá
HitTest	– Proces určovania
Middleware	– Počítačový softvér
NFL	– Národná futbalová liga
Open-source	– Bezplatná verzia
Off-line	– Neaktívny prenos
On-line	– Aktívny prenos
Outfit	– Oblečenie
Sceneform	– Nástroj pre vývoj rozšírenej reality
SDK	– Súbor nástrojov pre vývoj
Snapshot	– Snímka systému
UI	– Uživatelské rozhranie
WIFI	– Zariadenie pre pripojenie na bezdrôtovú lokálnu sieť
WWDC	– Celosvetové vývojárska konferencia
3D	– Trojrozmerný priestor

Zoznam obrázkov

2.1	The Sword of Damocle	11
2.2	Videoplace view	11
2.3	Virtual Fixtures	11
2.4	Dancing in Cyberspace	11
2.5	1st and Ten - žltá značka	11
2.6	NASA systém	12
2.7	Marta	13
2.8	Google Glass	13
2.9	Hololens	13
2.10	Ikea Place	13
3.1	Use-case diagram aplikácie	23
3.2	Návrh UI Welcome Page	24
3.3	Návrh UI hlavnej stránky	24
4.1	MyApp	26
4.2	Sekvenčný diagram	27
5.1	Zobrazenie UI na emulátore	30

Zoznam tabuliek

2.1	Tabuľka pre porovnanie AR frameworkov/SDK	17
-----	---	----

Kapitola 1

Úvod

Táto práca sa zaoberá veľmi zaujímavou oblasťou ktorou je rozšírená realita. Rozšírená realita tu s nami prebýva už niekoľko rokov, avšak ešte stále len prichádzame na množstvo užitočných vecí v ktorých je možné rozšírenú realitu využiť. „Rozšírená realita“ sa stále rozvíja a radí k neprebádaným oblastiam, ako aj umelá inteligencia, ktorá ešte len zažíva pomalý ale istý rozkvet v našom svete.

Pre väčšinu ľudí je pojem rozšírená realita stále neznámy. Taktiež si ju pletú s virtuálnou realitou, ktorá už je v dnešnom svete pomerne známa hlavne kvôli okuliarom pre virtuálnu realitu. Ja sama, som tento pojem nepoznala až po dobu, než sme s virtuálnou realitou a rozšírenou realitou mali možnosť pracovať na výmennom pobyte. To ma motivovalo k tomu, aby som si zvolila túto oblasť, ako tému mojej bakalárskej práce.

Dnes máme možnosť nájsť rozšírenú realitu vo viacerých odvetviach, ako sú aplikácie pre nábytok, hry, kinematografii, priemysle, vojsku, architektúre, obchode, literatúre alebo vo vizuálnom umení. Za posledných päť rokov sa viditeľne rozšírilo použitie rozšírenej reality. V súčasnej dobe vzniklo mnoho firiem a startupov, ktoré sa zameriavajú práve na zlepšovanie kvality života, využitím rozšírenej reality. Je len otázkou času kedy sa rozšírená realita stane neoddeliteľnou súčasťou každodenného života ľudí a firiem.

Cieľom našej práce je vypracovať aplikáciu pre rozšírenú realitu, ktorá umožní užívateľovi vkladať 3D obrázky do reálneho priestoru. Obrázky bude možné umiestňovať do priestoru, pohybovať s nimi a otáčať ich okolo svojej osi.

V teoretickej časti tejto práce sa zameriavame na postupný vývoj rozšírenej reality od roku 1968 až po dnešok. Veľkým prínosom bude aj jej využitie v budúcnosti. Budeme analyzovať nástroje pomocou ktorých, je možné vytvoriť aplikáciu s rozšírenou realitou, pre rôzne zariadenia a uľahčiť, alebo spríjemniť ľuďom ich každodenný život. Na základe tejto analýzy sme boli schopní určiť, vhodný nástroj pre náš vývoj. Tiež sme sa rozhodli uviesť na pravú mieru rozdiel medzi rozšírenou realitou a virtuálnou realitou. Aplikácia má pomôcť k lepšej predstavivosti, odhadu pri umiestnení objektov do priestoru interiéru.

Kapitola 2

Analýza Frameworkov

2.1 Rozdiel medzi frameworkom a SDK

Pred dvadsiatimi rokmi bolo možné spustiť aplikáciu z „command line“ a väčšinu kódu tvorila jedinečná logika programu. Dnes je potrebná veľmi komplexná podpora interaktívneho užívateľského rozhrania, správu siete a podobne.

V rámci Android frameworku [1] úlohou vývojára nieje stavať kompletný program, ale implementovať konkrétne správanie a potom ho správne zavolať z kostry frameworku. Framework je teda aplikácia alebo knižnica ktorá je pripravená k použitiu.

Android SDK, tiež nazývaný „devkit“ [2] zahŕňa sadu všetkých vývojárskych nástrojov a API, ktoré potrebuje vývojár pre pôsobivú a výkonnú mobilnú aplikáciu. V Android štúdiu je SDK dostupné pre každý operačný systém ako je napríklad Windows, MacOS, Linus. Najväčšou výzvou v SDK, ako aj v ďalších nových vývojárskych nástrojových sadách je naučiť sa vlastnosti a limity API. SDK je akoby jadrom API Androidu, ktorý zahŕňa všetko čo vývojár bude potrebovať na to aby mohol začať s vývojom, testovaním a ladením Android aplikácií. SDK je teda väčší koncept, pretože môže obsahovať knižnice, frameworky, dokumentáciu a nástroje.

2.2 Rozšírená realita/Predstavenie virtuálnej reality

Rozšírená realita (Augmented Reality / skratene AR) sa tiež zvykne nazývať aj sestrou virtuálnej reality. V čom sa odlišuje rozšírená realita od tej virtuálnej? Rozšírená realita je prepojenie reálneho sveta s virtuálnym. Na rozdiel od virtuálnej reality, ktorá vytvára úplne umelé prostredie, rozšírená realita využíva existujúce prostredie. To znamená, že rozšírená realita dokáže premietiť virtuálne objekty do reálneho sveta ktorý je okolo nás. Skutočný svet je teraz rozšírený resp. vylepšený o digitálne informácie. Tieto informácie sú získané pomocou rôznych zdrojov, za použitia on-line alebo off-line aplikácií. Obsah je možné zobraziť len za pomoci špeciálnych AR okuliarov alebo pomocou smartfónu/tabletu.

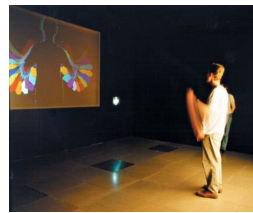
2.3 História a vývoj

Technológia rozšírenej reality bola vynájdená v roku 1968, keď Ivan Sutherland vyvinul prvý zobrazovací systém, ktorý sa pripevňoval na hlavu. Pojem „Rozšírená realita (Augmented reality)“ však vymyslel až výskumník zo spoločnosti Boeing, Tim Caudell v roku 1990.

V šesťdesiatych a sedemdesiatych rokoch Ivan Sutherland Harvardský profesor a počítačový vedec, vynášiel prvý displej pripevnený na hlavu nazývaný tiež „The Sword of Damocles“. Užívateľ mohol zažiť počítačom vytvorenú grafiku, ktorá zlepšila jeho zmyslové vnímanie sveta. Tento nástroj vyzeral ako na *Obrázku 2.1*.



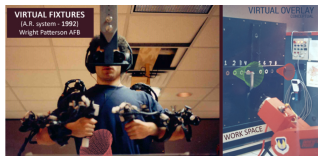
Obr. 2.1: The Sword of Damocle



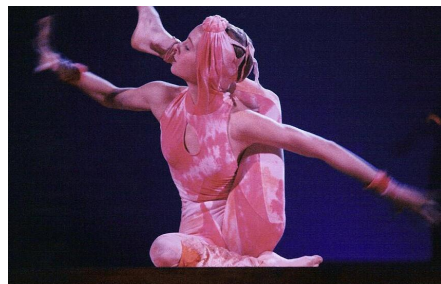
Obr. 2.2: Videoplace view

V roku 1974, Myron Kruger počítačový výskumník a umelec, vybudoval laboratórium na Univerzite v Connecticute s názvom Videoplace, ktoré sa venovalo iba umelej realite. V stenách bola použitá projekčná a kamerová technológia pre vyžarovanie siluet na obrazovke, ktoré obklopili používateľov kvôli interaktívnemu zážitku. To ako vyzerala táto miestnosť môžete vidieť na *Obrázku 2.2*.

V osemdesiatych a deväťdesiatych rokoch minulého storočia, konkrétne v roku 1992 Louis Rosenberg výskumný pracovník USA Armstrong's Research Lab, vytvoril program „Virtual Fixtures“, ktorý môžete vidieť na *Obrázku 2.3*. Bol jedným z prvých plne funkčných systémov rozšírenej reality. Tento systém umožňoval vojenskému personálu riadiť a viesť prakticky strojové zariadenia na vykonávanie úloh, ako je výcvik pilotov amerických vzdušných síl na bezpečnejšie letové praktiky.



Obr. 2.3: Virtual Fixtures



Obr. 2.4: Dancing in Cyberspace



Obr. 2.5: 1st and Ten - žltá značka

V 1994 sa rozšírená realita preniesla po prvý krát aj do zábavného priemyslu a to vďaka spisovateľke a producentke Julie Martin. Divadelnú produkciu s názvom *Dancing in Cyberspace* viz. *Obrázok 2.4*, predstavovala akrobatov tancujúcich popri premietaných virtuálnych objektoch na fyzickom pódiu.

V 1998 Sportsvision vysielala prvú živú hru NFL s virtuálnym grafickým systémom 1st and Ten - žltá značka, ktorá je zobrazená aj na *Obrázku 2.5*. Táto technológia zobrazuje žltú čiaru prekrývajúcu sa na vrchu informačného kanála. Zobrazenia pomôžu rýchlo zistiť, kam tím práve postupoval, aby získal priechod. Tento systém sa používa dodnes, hoci je nepochybne vyspelejší ako na konci 90. rokov. Diváci si zvykli na značku žltej čiary a ďalšiu doplnkovú grafiku - väčšina ani len netuší, že ide o formu technológie AR. [3] NASA v roku 1998 vytvorila hybridný systém syntetického



Obr. 2.6: NASA systém

videnia ich kozmickej lode X-38. Systém využíval technológiu AR na zabezpečenie lepšej navigácie počas testovacích letov. Komponent rozšírenej reality zobrazoval mapové údaje priamo na obrazovke pilota, ako môžete vidieť na *Obrázku 2.6*.

2.3.1 Rozšírená realita od roku 2000 a dnes

V roku 2000 Hirokazu Kato, vyvinul softvérovú knižnicu s otvoreným zdrojovým kódom s názvom ARToolKit. Tento balík pomáha ostatným vývojárom vytvárať softvérové programy pre rozšírenú realitu. Knižnica využíva sledovanie videa na prekrytie virtuálnej grafiky nad skutočný svet.

V 2003 Sportsvision vylepšila grafiku 1st and Ten tak, aby obsahovala funkciu nového systému Skycam - poskytla divákovi letecký záber poľa s grafikou prekrytou. V roku 2009 sa v *Časopis Esquire* prvýkrát použila rozšírená realita v tlačенých médiách pri pokuse o oživenie stránok.

MARTA (Mobile Augmented Reality Technical Assistance) viz. *Obrázok 2.7*, ktorá poskytovala technikom predovšetkým podrobné pokyny na opravu v servisnej príručke bola uvedená na trh spoločnosťou Volkswagen v 2013. Táto adaptácia technológie AR bola priekopnícka, pretože sa mohla a bude uplatňovať v mnohých rôznych odvetviach na zosúladenie a zefektívnenie procesov.



Obr. 2.7: Marta

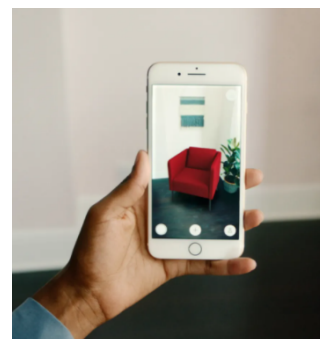
O rok na to Google predstavil svoje zariadenia Google Glass viz. *Obrázok 2.8*, dvojicu okuliarov s rozšírenou realitou, ktoré si môžu používatelia nosiť na pohlcujúce zážitky. Používatelia nosili technológiu AR a komunikovali s internetom pomocou príkazov na spracovanie prirodzeného jazyka. Pomocou tohto zariadenia mohli používatelia pristupovať k rôznym aplikáciám, ako sú Mapy Google, Google+, Gmail a ďalšie.



Obr. 2.8: Google Glass



Obr. 2.9: Hololens



Obr. 2.10: Ikea Place

V 2016 Microsoft začína dodávať svoju verziu technológie AR s názvom HoloLens viz. *Obrázku 2.9*, ktorá je pokročilejšia ako Google Glass, ale prišla na trh s nie príliš dostupnou cenou. Rozhodne to nie je každodenný typ príslušenstva. Headset pracuje na systéme Windows 10 a je v podstate prenosným počítačom. Používateľom tiež umožňuje skenovať ich okolie a vytvárať si vlastné zážitky z AR.

IKEA uviedla na trh svoju aplikáciu s rozšírenou realitou s názvom IKEA Place, ktorá navždy zmenila maloobchod.[3] Aplikácia umožňuje zákazníkom virtuálne zobrazíť ukážku ich možností domácej výzdoby pred skutočným nákupom. IKEA Place bola vyvinutá v 2017. Aplikácia umožňuje premietnuť 3D obrázky do reálneho sveta tak, ako môžete vidieť na *Obrázku 2.10*.

2.3.2 UNITY3D

Je multiplatformový veľmi populárny herný engine, vyvinutý spoločnosťou Unity Technologies. [4] Unity poskytuje možnosti vývoja pre 2D aj 3D hry ľubovoľného žánru a zamerania. Aj keď je Unity3D považovaný hlavne za nástroj pre vývoj hier, taktiež poskytuje mnoho riešení pre mobilnú rozšírenú realitu ako aj filmovú virtuálnu realitu. Unity3D tiež prichádza s Assets Store a trhom skriptov. Takže pomocou Unity3D môžete vytvoriť funkčnú aplikáciu bez potreby kódovania jediného riadku. Unity je spoplatnené, avšak existuje verzia pre študentov, ktorá je zdarma. Podporuje tvorbu skriptov C Sharp a JavaScript.[5]

2.3.3 AR CORE

ARCore [4] je framework na vytváranie aplikácií s rozšírenou realitou, vytvorená spoločnosťou Google. ARCore je možné použiť na vývoj pre Android aj iOS. Taktiež ARCore poskytuje obrovský výber rozhrania API. Poskytuje možnosť integrácie do Unity3D, Java, Android NDK, Ureal Engine a veľa ďalších. ARCore je úplná sada nástrojov pre multiplatformový vývoj. Je open-source nástroj. Pre ARCore sú charakteristické 3 kľúčové technológie. [6]

- Dokáže sledovať pohyb v reálnom čase
- Detekovať vzdialenosť a veľkosť objektov
- Lepšie odhadovať svetelné podmienky prostredia
- Web AR Core

2.3.4 VUFORIA

Vuforia [7] je ďalší populárny SDK pre vývoj priemyselných alebo podnikových, mobilných AR aplikácií. API je dostupné pre vývoj v C++, objektové C++, Java. Najviac populárne aplikácie ktoré boli vyvinuté Vuforiou sú Star Wars a Rabenity 2.0. Podporované platformy sú Android, iOS, UWP a Unity Editor. Vuforia nieje bezplatný SDK. Podporuje:

- Sledovanie cieľa
- Virtuálne tlačidlá
- Frame marker (Označenie rámu)

- Rozpoznávanie 2D a 3D cieľov
- Výber cieľového obrázka za behu
- Rozšírené sledovanie
- Smart Terrain TN
- Detekcia viacerých druhov cieľov (napr. Objektov, obrázkov, anglického textu)
- Aplikáciám tiež umožňuje používať na účely rozpoznávania obrázkov databázy, ktoré sú buď v lokálnom zdroji, alebo v cloude.

2.3.5 WIKITUDE

Je jeden z prvých vývojárskych balíčkov SDK pre Rozšírenú realitu, ktorý prišiel s kladovou integráciou. Wikitude [8] ako prvý začal pri vývoji aplikácie využívať prístup založený na polohe a bol ocenený ako najlepší vývojársky nástroj roku 2017. Wikitude je spoplatnený. Používa sa pri vývoji v Androide, iOS, Windows pre tablet, Smart Glasses. Wikitude SDK podporuje:

- Rozpoznávanie a sledovanie obrazu
- 3D vykresľovanie modelu a animácia
- Prekrytie videa
- Sledovanie podľa polohy (obrázok, text, tlačidlo a video)
- Rozšírenie HTML

2.3.6 EASY AR

EasyAR [9] je SDK ponúkaný v bezplatnej základnej verzii a v Pro verzii za licenčný poplatok 499 USD. EasyAR je tiež bezplatná alternatíva Vuforia a môže byť použitá pre širokú škálu vývoja pre Android, iOS, Windows. Ich riešenie používajú spoločnosti ako KFC, PepsiCo, Vogue a mnoho ďalších populárnych značiek. Funkcie EasyAR sú:

- Sledovanie plenárneho obrazu
- 3D sledovanie objektov s integráciou cloudu, pre zdieľané zážitky
- Ponúka tiež smart glass riešenie

2.3.7 KUDAN

Kudan SDK používa svoju patentovanú technológiu SLAM (Simultánna lokalizácia a mapovanie) na súčasnú lokalizáciu snímača vo vzťahu k svojmu okoliu, zatiaľ čo mapuje 3D štruktúru okolitého územia. SLAM je možné použiť s akýmkoľvek fotoaparátom alebo snímačom. Jej presnosť sledovania na fotoaparátoch smartfónov je presná až na milimeter. Vysoko-rýchlostný systém s nízkou spotrebou využíva menej ako päť percent procesoru mobilného zariadenia a naďalej funguje za extrémnych svetelných podmienok. Kudan ponúka vývojárom bezplatnú verziu svojej AR SDK na testovanie. Produkčná licencia stojí okolo 1000 USD ročne. Dostupný pre vývoj v Androide a iOS. [10]

2.3.8 ARToolkit+

Je [11] jeden z najstarších vývojárskych balíkov pre rozšírenú realitu z roku 1999. Bol vynájdený Washingtonskou univerzitou. ARToolkit+ je dostupný open-source framework. Má podporu rôznych platforiem ako sú Android, MS Windows, Linus, iOS, MAC OS X, a SGI. Prináša podporu OpenCV, OpenGL ES2.X, integráciu s GPS, automatickú kalibráciu kamery, ako aj integráciu s Unity3D. [4] Vlastnosti:

- Sledovanie orientácie
- 2D rozpoznávanie
- Lahká kalibrácia pre nové zariadenia
- Pomôcky na automatickú kalibráciu kamery
- Generovanie štvorcových značiek
- Obslužné programy na vytváranie značiek prirodzených prvkov

2.3.9 ARKit

Je framework vyvinutý spoločnosťou Apple. [4] Dostupný pre iOS vývoj. Bol navrhnutý tak, aby využíval hardvérové zariadenia ako iPhone/iPad. Na WWDC 2017 bol predstavený ako nástroj na vytváranie bezprecedentných natívnych aplikácií pre iPhone a iPad AR a na konkurenciu s inými technologickými spoločnosťami ako sú Facebook a Microsoft. ARKit [12] podporuje hlavne AR engine ako Unity, Vuforia, Unreal Engine. Úplne nový ARKit 4 predstavuje rozhranie Depth API, ktorý vytvára nový spôsob prístupu k podrobným hĺbkovým informáciám zhromaždeným skenerom LiDAR na zariadenia iPhone 12 Pro, iPhone 12 Pro Max a iPad Pro. Location Anchors využíva údaje s vyšším rozlíšením v aplikácii Apple Mapy na umiestňovanie AR zážitkov do konkrétneho bodu na svete.

	Unity3D	ARCore	Vuforia	Wikitude	EasyAR	Kudan	ARToolkit+	ARKit	AR.js
Typ	Framework	Framework	SDK	SDK	SDK	SDK	SDK	Framework	Framework
Cena licencie	Commercial	Open-Source	Open-Source, Commercial	Open-Source, Commercial	Open-Source, Commercial	Open-Source, Commercial	Open-Source	Open-Source	Open-Source
Programovacie jazyky	C#, Java, SWIFT, Objektové-C	Java, Kotlin	C#, C++	Java, Objektové-C	Java, Kotlin, SWIFT, Objektové-C	Java, Objektové-C	C++	Objektové-C, Swift	JavaScript
Platformy	Windows, Android, iOS	Android, iOS	Android, iOS, Windows	Android, iOS, UWP	Android, iOS, UWP, macOS	Android, iOS	Android, iOS, macOS, Linux, Windows	iOS	WebGL
Smart Glasses	●	●	●	●	○	○	●	●	○
Cloudové rozpoznávanie	●	○	●	●	●	○	●	○	●
Integrácia s Unity3D	●	●	●	●	●	●	●	●	○
SLAM	●	○	○	●	●	●	●	●	○
3D rozpoznávanie	●	●	●	●	●	●	●	●	●

Tabuľka 2.1: Tabuľka pre porovnanie AR frameworkov/SDK

2.3.10 AR.js

Tento framework povoľuje sprístupňovať aplikácie rozšírenej reality priamo z prehliadača. Je to verzia programu ARToolkit špeciálne upravená pre webové prehliadače. Vývojový tím sľubuje kompatibilitu a vysoký výkon vo všetkých vyhľadávačoch, ktoré používajú WebGL, a WebRTC. Hneď po niekoľkých riadkoch kódu HTML je k dispozícii funkčná ukážka aplikácie AR. AR.js je open-source produkt, ktorý je zadarmo na použitie pre každého. Výhody sú napríklad také, že nieje potrebné inštalovať aplikáciu. Používateľom na dosiahnutie obsahu stačí naskenovať AR kód pomocou fotoaparátu telefónu.[13]

2.3.11 Porovnanie a zhrnutie

Podľa môjho názoru je pre vývoj rozšírenej reality najlepšie použiť buď Unity3D, ARKit alebo ARCore. Samozrejme že všetko závisí aj od toho či chcete vyvíjať rozšírenú realitu pre Android alebo iOS. Rozhodla som sa použiť ARCore a SDK Sceneform, pretože majú zrozumiteľnú a dostupnú dokumentáciu, tutoriály a iné zdroje ktoré je možné použiť na vývoj aplikácii pre rozšírenú realitu. Ja vo svojej práci používam konkrétne ARCore a Sceneform SDK, pretože sa mi páči funkčnosť, vlastnosti frameworku a integrácia s rôznymi API. [14]

V tabuľke 2.1 môžete vidieť zhrnutie a porovnanie frameworkov/SDK pre vývoj rozšírenej reality.

2.3.12 Budúcnosť rozšírenej reality

Pravdou je, že rozšírenú realitu už používajú každodenní spotrebitelia - len o tom nevedia. Veľmi rozšíreným príkladom, pre každodenné používanie tejto reality, sú napríklad filtre zakomponované v sociálnych sieťach ako Snapchat, Messenger, Instagram.

Psí filter na Snapchat je jeden z príkladov, ktorý pozná určite každý z nás. Najväčším posunom v rozšírenej realite bude musieť byť spôsob, akým sa zmení jej vnímanie. Oproti virtuálnej realite je rozšírená realita viac zameraná na každodenné používanie. [4]

Už dnes pokrýva veľkú časť využitia. Avšak v budúcnosti si napríklad bude možné pomocou rozšírenej reality urýchliť výber denného outfitu z kúskov, ktoré bude náš šatník obsahovať, pretože si ho pomocou rozšírenej reality budeme môcť kedykoľvek poskladať a pripraviť dané diely oblečenia o deň skôr. V tom prípade nebudeme strácať drahocenný čas ktorí strávime každý deň rozhodovaním v šatníku. Tiež nebude hroziť že zabudnete na kúsky v šatníku.

Veľký potenciál určite vidím aj v spojení rozšírenej reality s umelou inteligenciou. Napríklad pri spomínanom šatníku, bude umelá inteligencia rozhodovať či sa daný outfit hodí na zadanú udalosť, kde treba aby bol splnený určitý kódex a podľa toho nám budú ponúknuté možnosti. Vyhneme sa tak nesplnenému kódexu obliekania.

2.4 AR Core a Sceneform SDK

ARCore je platforma od Googlu pre budovanie zážitkov z rozšírenej reality pomocou rôznych API. ARCore umožňuje telefónu, vnímať prostredie, porozumieť svetu a pracovať s informáciami. Existujú tri hlavné komponenty pre správne fungovanie ARCore, ktoré už boli v krátkosti spomenuté v predchádzajúcej kapitole.

Prvou z nich je Motion Tracking (Sledovanie pohybu), Environmental understanding (Porozumenie prostrediu) a Light estimation (Odhad svetla). Motion tracking využíva vizuálny vstup z kamery, na detekovanie jedinečných vlastností v prostredí. Využíva pre to polohu a orientáciu kamery prostredníctvom vstupov zo senzorov telefónu, ako sú akcelerometer a gyroskop.

To spočiatku zaručuje virtuálny obsah vzhľadom na aktuálnu pozíciu užívateľa. Sensory ako akcelerometer a gyroskop ďalej zvyšujú schopnosť porozumieť pohybu užívateľa v prostredí. To znamená, že napríklad ste vytvorili 3D objekt v kuchyni, prechádzali ste kolo objektu a potom ste sa vrátili späť. Kamera v tomto prípade bude schopná porozumieť kde bol daný objekt umiestnený a preto ho umiestni na to isté miesto. [15]

ARCore je tiež inteligentná v tom zmysle, že dokáže porozumieť prostrediu. Napríklad keď vidí veľa prvkov označených vodorovne alebo zvisle, chápe že existuje horizontálna alebo zvislá čiara v tejto oblasti. Toto napomáha umiestneniu objektu pomocou ARCore. Taktiež osvetľuje predmety za rovnakých podmienok aké poskytuje prostredie. Pomáha tiež s korekciou farieb a úpravou intenzity svetla, ak je to potrebné.

Sceneform SDK je dá sa povedať nadstavba pre ARCore. Pomocou Sceneform SDK je možné importovať a upravovať 3D modely vo formáte (.obj, .fbx, .glTF) a na základe toho vytvárať realistickú 3D scénu pre ARCore aplikácie, bez potreby učenia sa OpenGL alebo OpenCV.

Štandardne spracováva pre nás množstvo užívateľských skúseností, ako napríklad požiadanie užívateľa o skenovanie prostredia a následné umiestnenie 3D objektu do prostredia skutočného sveta.

Ťuknutím na obrazovku telefónu budeme schopní vykonať konverziu, ktorá je v 2D súradniciach na obrazovke a v 3D v skutočnom svete. To sa deje pomocou API s názvom HitTest, ktoré poskytuje Google. To čo robí HitTest, je v podstate klepnutie na telefón a jeho prevedenie na niečo ako lúč svetla, ktorá ide po priamke a následne sa pretne s reálnym svetom.[6]

2.4.1 Augmented Images (Rozšírené obrázky)

AR rozšírených obrázkov v ARCore umožňuje vytvárať aplikácie AR, ktoré dokážu detekovať a zväčšiť 2D obrázky v prostredí používateľa, napríklad plagáty alebo obaly produktov. ARCore dokáže sledovať až 20 obrázkov súčasne. ARCore nebude súčasne zisťovať ani sledovať viac inštancií toho istého obrázku. Každá databáza rozšírených obrázkov môže obsahovať informácie až tisíceky referenčných obrázkov. Počet databáz nieje obmedzený.

Pri pridávaní obrázkov je možné určiť fyzickú veľkosť obrázka. Týmto krokom sa zlepši výkon detekcie obrazu. Ak nieje poskytnutá žiadna fyzická veľkosť, ARCore odhadne veľkosť a tento odhad v priebehu času spresní. Ak je poskytnutá fyzická veľkosť ARCore použije zadanú veľkosť a odhadne pozíciu a orientáciu obrazu, pričom ignoruje akýkoľvek nesúlad medzi zjavnou alebo skutočnosťou veľkosťou a poskytnutou fyzickou veľkosťou.

ARCore dokáže reagovať a sledovať obrázky, ktoré sú: upevnené na svojom mieste, napríklad tlač zavesená na stene alebo časopis na stole, pohyblivé obrázky ako napríklad reklama na okoloidúcom autobuse alebo obrázok na plochom predmete, ktorý drží užívateľ pri pochybe rukami. Akonáhle ARCore začne sledovať obraz, poskytuje odhady polohy a orientácie obrazu pre každú snímku. ARCore tieto odhady neustále zdokonaľuje, pretože zhromažďuje viac údajov.

Akonáhle je obraz hotový ARCore pokračuje v „sledovaní“ polohy a orientácie obrazu, a to aj v prípade, že sa obraz dočasne presunie z dohľadu kamery, pretože užívateľ presunul svoje zariadenie. Pri tom ARCore predpokladá, že poloha a orientácia obrazu sú statické a samotný obraz sa nepohybuje prostredím. [15]

Kapitola 3

Analýza a návrh aplikácie

V predchádzajúcej kapitole sme vykonali prieskum dostupných knižníc pre vývoj aplikácie rozšírenej reality. Z toho sme zistili že existuje množstvo knižníc ktoré je možné použiť a kombinovať pri vývoji našej aplikácie.

V tejto kapitole sa budeme zaoberať hlbšou analýzou aplikácie. Vďaka analýze bude možné vybrať správny nástroj pre vývoj. Porovnáme a zhrnieme dosiahnuté výsledky s inými dostupnými produktami na trhu. Budeme sa tiež venovať rolám užívateľa, ako aj tokom ktoré sú s týmito rolami spojené. Role budú zobrazené pomocou príslušného v UML diagramu. Návrh, ako aj to prečo sa návrh používa, bude ďalšou témou, ktorou sa budeme v tejto kapitole zaoberať.

3.1 Požiadavky na aplikáciu

V prvom rade, bolo dôležité ujasniť si požiadavky kladené na vývoj softvéru. Naším cieľom je vytvoriť aplikáciu, ktorá je plne využiteľná pre Android a bude blízka užívateľom po grafickej stránke a užívateľskej stránke. Aj k tomu nám dopomohol návrh, ktorý sme si vopred zhotovili.

V aplikácii bude zahrnuté vkladanie 3D obrázkov z internetového úložiska, bez potreby upravovania kódu, ako aj priame vystavanie obrázku zadaním url adresy do aplikácie. Aplikácia bude schopná premietiť 3D obrázky do skutočného sveta. Taktiež, bude možné tieto obrázky otáčať a pohybovať s nimi po miestnosti. V niektorých prípadoch je 3D obrázky možné zmenšovať a zväčšovať do určitej veľkosti. Toto závisí od veľkosti a formátu súradníc daného obrázku.

Užívateľ bude mať možnosť tiež navrhnuté rozloženie obrázkov v priestore interiéru uložiť do galérie fyzického zariadenia pomocou funkcie snímku obrazovky, pre neskoršie použitie. Medzi požiadavkami tiež bolo potrebné výslednú aplikáciu otestovať na viacerých typoch hardvérových a softvérových zariadení. Testovaniu sa budeme venovať v poslednej kapitole našej bakalárskej práce.

3.2 Role

V našej aplikácii bude jeden typ užívateľa/role ktorý bude oprávnený používať našu aplikáciu.

Užívateľ bude schopný si danú aplikáciu stiahnuť, nainštalovať a pomocou nej premietat 3D obrázky do priestoru interiéru v ktorom sa nachádza. S obrázkami bude môcť užívateľ aplikácie pohybovať po priestore a otáčať ich. Tiež je možné určité typy obrázkov zväčšovať alebo zmenšovať. 3D obrázky v podobe nástennej maľby je možné umiestniť do vyhovujúcej polohy na stenu. Užívateľ bude môcť pracovať s galériou, pridávaním 3D obrázkov do aplikácie cez internetové úložisko na Firebase. 3D obrázky bude môcť ukladať pomocou URL adresy. Užívateľ bude schopný uložiť náhľad plochy obrázku pomocou zachytenia plochy, ktoré sa následne uložia do galérie telefónu. Na základe toho bude možné vrátiť sa k náhľadu neskôr a to bez potreby internetového pripojenia.

3.3 Use-Case diagram

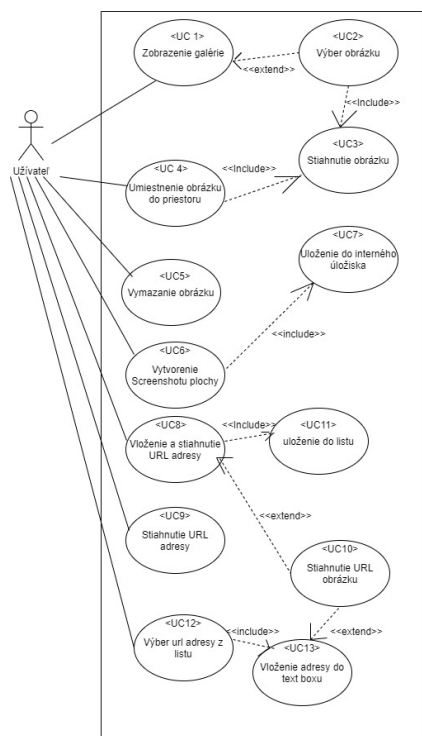
V tejto kapitole si pomocou Use-Case diagramu vid. *Obrázok 3.1* vymedzíme role a funkcie užívateľa, ktoré sme definovali vyššie. Use-Case diagram alebo tiež známi diagram prípadov užitia, zachycuje vonkajší model systému, ktorý slúži ako podklad pre odhady rozsahu systému. Naš nasledujúci diagram prípadov užitia zachytáva to, s čím má možnosť pracovať užívateľ pri používaní našej aplikácie. Ako napríklad možnosť stiahnutie a umiestnenie 3D obrázku na scénu zariadenia snímajúceho priestor, alebo vymazanie 3D obrázkov zo scény.

3.4 Voľba nástroja pre vývoj

Pred začatím vývoja našej aplikácie sme sa museli rozhodnúť, ktorý nástroj použijeme pri vývoji. Zvolenie správneho nástroja, je veľmi dôležité pretože, tým zabránime prípadným komplikáciám, ktoré by sa mohli počas vývoja objaviť. Do úvahy prichádzali dve možnosti frameworkov pre rozšírenú realitu:

- prvým z nich bol Unity3D ktorý sa často používa pri vývoji multiplatformovej aplikácie s využitím virtuálnej alebo rozšírenej reality (táto aplikácia by bola vyvíjaná priamo v programe od Unity3D za využitia skriptovacieho jazyka JavaScript)
- druhou možnosťou bola už spomínaná ARCore s využitím SDK Sceneform (túto aplikáciu bolo možné vyvíjať v Android Studiu využitím programovacieho, objektovo orientovaného jazyka Javy)

Pre tento typ aplikácie, ktorá umiestňuje 3D objekty do interiéru sme sa rozhodli použiť druhú možnosť v poradí. ARCore umožňuje použiť všetky spomínané funkcie, ktoré sme sa do našej aplikácie snažili zakomponovať. Dôvodom bolo tiež využitie Android Studia a Java jazyka, ktorý je v súčasnosti populárny a v rebríčkoch radený medzi najviac využívané programovacie jazyky. Taktiež



Obr. 3.1: Use-case diagram aplikácie

sme sa riadili aj osobnými preferenciami. Samotný softvérový rámec ARCore od Googlu, má dobre prepracovanú dokumentáciu a možnosti integrácie s inými API. Dokumentácia bola veľmi silným faktorom pri výbere softvérového rámca.

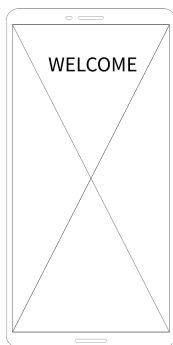
3.5 Prvotný návrh UI

V nasledujúcej podkapitole sa budeme venovať návrhu grafického rozhrania aplikácie a tiež tomu, prečo je vhodné vytvoriť návrh užívateľského rozhrania aplikácie.

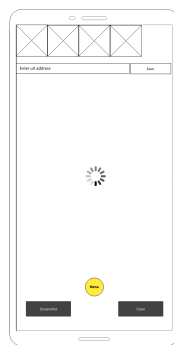
Dizajnerský návrh je dokument alebo rad snímok, ktoré stanovujú plán dizajnerskeho projektu. Vďaka návrhu UI je možné rýchlejšie získať klienta, ktorému sa vizualizovaný návrh môže zapáčiť. Hlavným zámerom dizajnu a architektúry UI je jednoduchosť a intuitívnosť používania aplikácie užívateľom.[16]

Na týchto *Obrázkoch 3.3, 3.2* môžete vidieť prvé návrhy vzhľadu užívateľského rozhrania našej aplikácie. Na obrázku vľavo sa nachádza uvítacia stránka, ktorá sa zobrazí na pár sekúnd hneď po otvorení aplikácie.

Po zobrazení úvodnej stránky sa zobrazí hlavná stránka viz. *Obrázok 3.2* našej aplikácie s ktorou bude užívateľ pracovať po celú dobu používania aplikácie. Kladený bude dôraz na využívanie ArFragmentu, pre premietanie 3D obrázkov do priestoru reálneho sveta. Ten je možné do súboru Android layoutu pridať ako každý fragment systému Android. Fragment kontroluje, či je nainštalovaný.



Obr. 3.2: Návrh UI Welcome Page



Obr. 3.3: Návrh UI hlavnej stránky

vaná verzia ARCore kompatibilná, a vyzve používateľa, aby si ARCore podľa potreby nainštaloval alebo aktualizoval. Fragment navyše určuje, či má aplikácia prístup ku kamere, a požiada používateľa o povolenie, ak ešte nebolo udelené. Fragment tiež vytvára ArSceneView a ARCore Session čo znamená že scéna a relácia sú spracovávané automaticky. [17]

V prípade hlavnej stránky budeme používať ConstraintLayout ktorý je najvhodnejší. Jedným z používaných dizajnerských vzorov v našej aplikácii bude tiež RecyclerView, ktorý používame pre vytváranie náhľadov obrázkov v galérii.

Kapitola 4

Implementácia aplikácie

Podľa zadania bola vypracovaná Android aplikácia pre vizualizáciu interiéru za pomoci rozšírenej reality. Aplikácia je schopná umiestňovať obrázky rozšírenej reality do miestnosti, taktiež je možné s objektami pohybovať, premiestňovať ich, zväčšovať alebo nimi otáčať. Cieľom aplikácie je čo najrealistickejšie umiestniť daný objekt do priestoru.

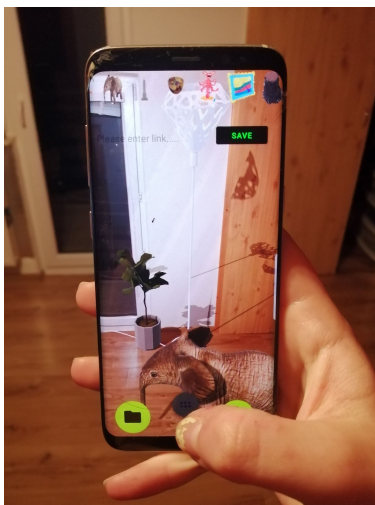
Pre implementáciu bol použitý softvérový rámec ARCore a plugin SDK Sceneform pre spracovanie, zobrazovanie a budovanie 3D obrázkov. SDK Sceneform je bližšie popísaný v predchádzajúcej kapitole 4 ARCore a SDK Sceneform. Naša výsledná aplikácia by mala slúžiť pre lepšiu vizualizáciu Interiérových doplnkov. V nasledujúcej sekcii sa budeme zaoberať postupom, ktorý bol zvolený pri vývoji aplikácie. Implementácií predchádzalo zhrnutie informácií potrebných pre vývoj danej aplikácie ako aj výber správnych vývojárskych nástrojov. Na *Obrázoku 4.1* môžete vidieť výsledné použitie našej aplikácie.

4.1 Použité knižnice

Nami použité knižnice sú kompatibilné s platformou Android. Vďaka použitej knižnici ARCore a Sceneform pre vytváranie realistickej 3D scény, nebolo potrebné učiť sa OpenGL. Sceneform automaticky importuje 3D modely a potom ich asynchrónne načíta a spracuje. Použité knižnice sme predstavili a rozobrali v predchádzajúcej kapitole.

4.2 Technické požiadavky

Kvôli vyššej náročnosti aplikácie na procesor telefónu, je možné aplikáciu stiahnuť a používať len zariadeniach podporujúce OpenGL ES 3.0 alebo OpenGL ES 3.2. Tiež je požadovaná verzia Androidu 8.0 a vyššia.



Obr. 4.1: MyApp

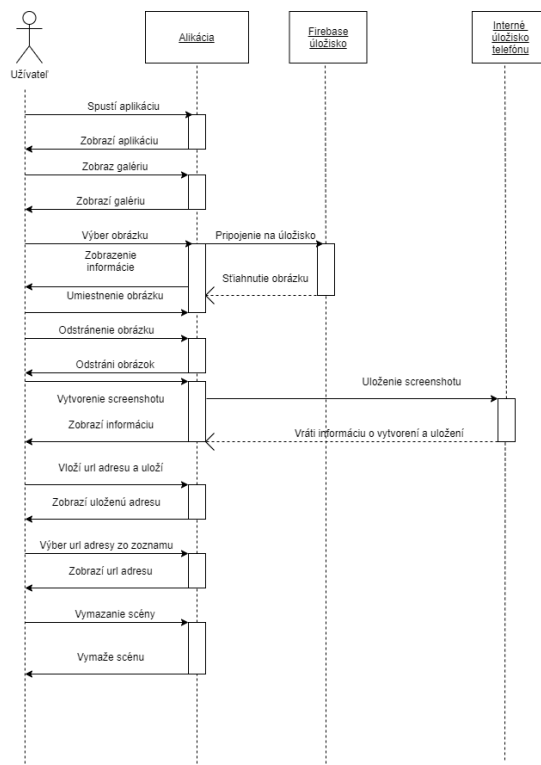
4.3 Vývoj aplikácie

Po analýze a následnom zvolení nástroja sa dostávame do fáze kedy sme zahájili vývoj. Najskôr bolo potrebné pripojiť do Android Studia nástroje pre vývoj rozšírenej reality, Sceneform plugin a ARCore. Sceneform zahŕňa rozhranie API grafov na vysokej úrovni, realistické fyzicky založené vykresľovacie zariadenie, doplnok Android Studia na import, prezeranie, vytváranie 3D prvkov a ľahkú integráciu do ARCore, ktorá umožňuje jednoduché utváranie aplikácií AR. Začali sme s vývojom. Bolo potrebné zoznámiť sa s prostredím a zistiť možnosť ukladania a zobrazovania 3D obrázku do priestoru pomocou telefónu. Následne sme boli schopný vytvoriť galérie s obrázkami a implementovať ostatné funkcie. Na konci vývoja aplikácie máme užívateľské rozhranie, ktoré sa viacmenej podobné návrhu.

Na sekvenčnom diagrame viz. *Obrázok 4.2* sú zachytené postupy, ktoré sa dejú v aplikácii počas používania užívateľom. Môžeme na ňom vidieť napríklad sekvenciu správ, ktoré posielajú medzi sebou užívateľ a aplikácia. Taktiež tu môžeme vidieť postup pri ukladaní adres obrázkov, ako aj postup na zobrazovanie 3D obrázkov.

4.4 Pripojenie

Pripojenie našej aplikácie k úložisku bolo realizované prostredníctvom realtime databázy na Firebase. Pre prácu s touto aplikáciou musí byť užívateľ v online režime telefónu. Ak užívateľ nieje v online režime aplikácia ho následne upozorní na potrebu pripojenia. Práca s aplikáciou je podmienená pripojením na internet, počas celej doby používania. Sťahovanie obrázkov pre túto aplikáciu bolo realizované cez spomínané úložisko Firebase od spoločnosti Google. Na základe tohto pripoje-



Obr. 4.2: Sekvenčný diagram

nia, zobrazeného v *Ukážke kódu 4.1*, bolo potrebné vytvoriť pripojenie cez závislosti k Firebase. Pre ukladanie obrázkov plochy, aplikácia využíva interné úložisko, mobilného telefónu užívateľa.

```
implementation 'com.google.firebase:firebase-database'
```

Listing 4.1: Závislosti pre pripojenie k databázi

Neskôr pri rozširovaní funkcionality sme tiež využili sťahovanie obrázku priamo z internetu a jeho ladenie, cez URL adresu v aplikácii. Tiež bolo využité aj lokálne úložisko aplikácie. Pre ukladanie na lokálne úložisko sme zvolili Shared Preferences namiesto SQLite, pretože nebolo potrebné pracovať s veľkým množstvom dát ako je to pri SQLite. Dáta zo Shared Preferences je tiež možné načítať späť.

4.5 Renderovanie 3D obrázku

V tejto kapitole sa budeme bližšie venovať metódam a funkciám ktoré boli použité pri vývoji. Po správnom načítaní obrázkov do galérie aplikácie prichádza rad na renderovanie a umiestňovanie 3D obrázkov v priestore. Pre tento účel sme vytvorili nasledujúce metódy.

Metóda `BuildARModelAsset()`

Metóda umožňuje vystavať 3D obrázok. Pre funkcionality tejto metódy je dôležité zmieniť Rende-

renderableSource, ktorý môžete vidieť použitý v *Ukážke kódu 4.2*. S jeho pomocou je možné nakonfigurovať a vytvoriť prevodník. ModelRendererable zobrazený v *Ukážke kódu 4.3*, poskytuje asynchrónnu implementáciu procesu načítania, teda ide o to, že po dokončení načítania môžeme pokračovať v pridávaní uzla obrázkov na scénu.

```
RenderableSource renderableSource = RenderableSource
    .builder()
```

Listing 4.2: Metóda pre vystavanie 3D obrázku

```
ModelRenderable.builder()
```

Listing 4.3: Metóda pre vystavanie 3D obrázku

Metóda setUpPlane()

Pomocou tejto metódy viz. *Ukážku kódu 4.4*, je možné umiestniť 3D objekt do priestoru. Transformable node slúži k tomu aby sme boli schopný s 3D objektom pohybovať po priestore alebo ho otáčať. AnchorNode slúži na vytváranie získavanie a ukotvenie 3D obrázkov v priestore. Pomocou ScaleControlleru je možné objektu nastaviť realistickejší vzhľad. LocalPosition a LocalRotation sa používa pre zmenu pozície vo vzťahu k rovine, napríklad ak máme luster, ktorý by mal byť umiestnený zvislo na strop, tieto metódy umožňujú pracovať s lustrom aj vo zvislých rovinách. Výsledkom toho bude zvislo zobrazený luster visiaci zo stropu nadol.

```
AnchorNode anchorNode = new AnchorNode(hitResult.createAnchor());
TransformableNode node = new TransformableNode(arFragment.
    getTransformationSystem());
node.getScaleController().setMinScale(0.3999f);
node.getScaleController().setMaxScale(0.4000f);
node.setLocalPosition(new Vector3(0, 0, 0));
node.setLocalRotation(new Quaternion(0, 0, 1, 0));
```

Listing 4.4: Metóda pre umiestnenie 3D objekt do priestoru

Kapitola 5

Testovanie

Aplikáciu sme testovali na zariadeniach so systémom Android. Testovanie bolo zrealizované na zariadeniach, Samsung:

- Samsung Galaxy S8, podporuje viac rozlíšení textúr GPU, GPU - ARM Mali-G71, Depth API, verzia operačného systému Android 9.0, OpenGL ES 3.2, Displej - 2960 x 1440, 5.8
- Samsung Galaxy A51, GPU - ARM Mali-G72, Depth API, verzia operačného systému Android 11.0, OpenGL ES 3.2, Displej - 2400 x 1080, 6.5

A taktiež na Android emulátore:

- Pixel 4 API 29, Android 10.0, x86, OpenGL ES 3.1

Pričom emulátor mal nastavený jazyk systému na Anglický. Fyzické zariadenia mali nastavený Slovenský jazyk.

Otestovať našu aplikácie bolo možné viacerými spôsobmi. Prvá možnosť, ktorá sa nám ponúkala, bolo testovanie aplikácie na fyzickom zariadení. Na týchto zariadeniach fungovali funkcie aplikácie bezchybne. U týchto typov zariadení bola veľmi rýchla odozva, na zadanú úlohu. Avšak u Samsung Galaxy A51 sme si všimli, že ak napríklad s aplikáciou pracujeme dlhšie, tak po nejakom čase sa aplikácia sama vypne a opäť naštartuje. Problém nastal pri sledovaní scény, kamera telefónu prestala snímať okolie.

Druhým je testovanie pomocou emulátora, ktorý ponúka každý mobilný softvérový rámec. Keď sme našu aplikáciu rozbiehali na emulátore konkrétne na Pixel 4 API 29, užívateľské rozhranie zostalo nezmenené. Pri emulátore v Android Studiu, sme si všimli, že má nepostačujúcu rýchlosť. Pri prevádzaní operácií trvalo hodnú chvíľu kým sa daná operácia uskutočnila. Pre spustenie na Android Emulátore bolo nutné stiahnuť a nainštalovať najnovšiu verziu ARCore APK. Zobrazenie na emulátore môžete vidieť tu 5.1.



Obr. 5.1: Zobrazenie UI na emulátore

5.1 Nasadenie aplikácie

Pri nasadení našej aplikácie sme používali Android Studio. Pre spustenie aplikácie a sťahovanie 3D obrázkov z internetového úložiska bolo potrebné Wifi pripojenie. Naša aplikácia si vyžaduje internetové pripojenie. Pre sťahovanie obrázkov z Firebase databázy, je možné vytvoriť si vlastnú databázu, galériu obrázkov.

Firebase je sada nástrojov na zostavenie, vylepšenie a rozšírenie vašej aplikácie, pokrývajúce veľkú časť služieb, ktoré by si vývojári museli zvyčajne sami vyskladať. Patria sem napríklad analýtika, autentifikácia, databázy, konfigurácia, ukladanie súborov, alebo zasielanie správ.

Služby sú hostované v cloude a sú škálovateľné s minimálnym alebo žiadnym úsilím zo strany vývojára. Klientske súpravy SDK poskytované serverom Firebase interagujú s týmito backendovými službami priamo, bez toho, aby ste medzi vašou aplikáciou a službou museli vytvárať nejaký middleware. Ak teda používate niektorú z možností databázy Firebase, zvyčajne napíšete kód na vyhľadanie databázy v klientskej aplikácii.[18]

5.2 Testovanie pri vývoji

Priebežné testovanie našej aplikácie malo pomôcť predísť neželaným chybám. Bola testovaná na viacerých zariadeniach, kde sme odchyťovali prípadné nedostatky v užívateľskom rozhraní, a funkcionality. S pribúdajúcim časom sa nám na aplikácii začali objavovať jemné rozdiely na zariadeniach.

Tieto nepresnosti sa týkali skôr správania sa dizajnu na rôznych zariadeniach Androidu či emulátoru. Na základe toho sme dokázali dizajn v aplikácii správne vyladiť. Na funkcionality aplikácie zmena zariadenia vplyv nemala.

5.3 Priebeh testovania na platforme Android

Po spustení aplikácie sa nám ako prvá zobrazí uvítacia aktivita s jednoduchým návodom. V našom prípade pri fyzickom zariadení sa jazyk aplikácie nastavil na Slovenský. Aplikáciu sme následne otestovali aj s nastaveným Anglickým, Ruským a Nemeckým systémovým jazykom. V prípade ak bolo na zariadení vypnuté internetové pripojenie, aplikácia upozorní užívateľa na potrebu zapnutia mobilných dát, alebo Wifi.

Po zapnutí mobilných dát je možné s aplikáciou pracovať. Kliknutím na galériu, a vybratím kategórie sa nám načítajú údaje z databázy. Tieto údaje v databáze je možné meniť a upravovať, aj za chodu aplikácia. Databáza je spoločná pre všetky zariadenia. Po výbere obrázku sa obrázok začne sťahovať a vytvárať. Na určenie stavu obrázku nám pomáha progressbar, ktorý sa zapne pri sťahovaní obrázku a vypne hneď po tom, čo je obrázok pripravený k umiestneniu na scénu.

Keď je obrázok umiestnený na scéne, je možné s nim pohybovať po priestore, v niektorých prípadoch ho zväčšovať alebo zmenšovať, tiež je možné s nim rotovať po priestore. Na scénu je možné pridať viac objektov. Tiež je možné celú hierarchiu objektov vymazať. Screenshot (snímka obrazovky), ktorý využíva Pixel Copy, pre zachytenie 3D obrázkov v reálnom svete, je možné stiahnuť a umiestniť do galérie telefónu. Užívateľ tak má možnosť pristupovať k vytvoreným dizajnom aj v off-line režime.

5.4 Užívateľské testovanie

Testovanie už finálnej verzie aplikácie nám pomôže predísť problémom a potvrdiť funkčnosť aplikácie pre konečného zákazníka. Automatické testovanie tiež známe ako regresné testovanie je rýchle, spoľahlivé. Tento typ testovania je skôr bežný pre vývojárov pracujúcich na veľkých informačných systémoch.

Druhou možnosťou je manuálne testovanie. Manuálne testovanie sa od toho automatického líši tým, že je časovo náročnejšie. Keďže pri manuálnom testovaní zohráva veľkú úlohu aj človek, ako užívateľ, pretože aj užívateľovi sa môže stať, že prehliadne chyby, je toto testovanie nákladné a nespoľahlivé. Ako aplikácia rastie, rastú nám aj náklady na manuálne testovanie.¹

Tento druh testovania sme zvolili pri testovaní aj my. Z pozície vývojára môžeme usúdiť, že zatiaľ čo vývojár si všimá zložitejších problémov, ktoré nastali pri testovaní, typický užívateľ sa pozerá na aplikáciu komplexnejšie.

¹<https://www.learn2code.sk/blog/manualne-vs-automatizovane-testovanie>

5.5 Konečné hodnotenie testovania

Čo sa týka samotnej platformy ARCore, nemám k nej žiadne výhrady. Simultánna lokalizácia a mapovanie priestoru ktorý ARCore využíva na to aby zariadenie pochopilo v akom prostredí sa nachádza, bolo bezchybné, vždy vedelo rozoznať vodorovné a zvislé plochy scény. Na základe tohto mapovania bolo jednoduché premiestňovať súradnice obrázku po priestore, alebo s nimi inak ďalej pracovať. Obrázok, akoby stal súčasťou skutočného sveta. Toto všetko pomáhalo vykresľovaniu 3D zo správnej perspektívy.

Čo sa týka odhadu svetla, zistením tejto informácie sme mali možnosť osvetliť virtuálne objekty za rovnakých podmienok ako prostredie v ich okolí. Toto zvýšilo ich zmysel pre realitu a dodalo iný nádych virtuálnemu objektu.

Je však potrebné aby sme spomenuli že pri detekovaní napríklad bielej steny bez textúr, sme spozorovali určité nedostatky. ARCore nie vždy bola schopná zachytiť uhol povrchu bielej steny. Toto bolo jediné zistenie, ktoré sme spozorovali pri testovaní softvérového rámca ARCore.

ARCore hodnotíme ako prepracovaný framework s množstvom možností. Dúfame, že v budúcnosti bude, možné rozšírenú realitu využívať viac účelovo. Aplikácie s ktorými sme sa mali možnosť doteraz stretnúť boli skôr, zamerané na zábavu ako úžitok. Aplikácia od IKEA Place, ktorá bola pre nás inšpiráciou v našom výbere bakalárskej práce. Táto aplikácia od IKEI je jedna z mála aplikácií, ktoré ponúkajú informatívny význam. Avšak v porovnaní s našou aplikáciou, IKEA Place neumožňuje užívateľovi pridávať vlastné objekty.

Kapitola 6

Záver

Na začiatku bolo potrebné zoznámiť sa s pojmom rozšírená realita. Potrebovali sme si stanoviť presný cieľ vývoja našej aplikácie. Cieľom práce bolo vytvorenie funkčnej mobilnej aplikácie, pre umiestňovanie 3D objektov do priestoru interiéru.

Bolo nutné vybrať a zistiť ako pracujú jednotlivé knižnice pre vývoj rozšírenej reality. Začiatočná nevedomosť vo vývoji tejto aplikácie sa zmenila. Práca na projekte bola veľmi užitočná. Vidím stále nedostatky v mojich vývojárskych schopnostiach a zručnostiach, no aj vďaka tejto aplikácii sa mi ich podarilo zlepšiť. Som veľmi rada, že som mala mohla vyvíjať aplikáciu pre Android, pretože pri samotnom štúdiu som nikdy nemala možnosť túto šancu využiť. Naša aplikácia bola zameraná na umiestňovanie objektov na rôzne povrchy v priestore.

Na základe vykonaného prieskumu sme zistili, ktorý softvérový rámec bude pre našu aplikáciu najvhodnejší. Pomocou ARCore a SDK Sceneform sa nám podarilo vytvoriť aplikáciu, ktorá dokázala prenášať 3D objekty do reálneho prostredia. Následne bolo možné s daným objektom ďalej pracovať, premiestňovať ho po priestore, otáčať do potrebnej polohy, zväčšovať. Bližšie sme popísali framework ARCore, SDK Sceneform a iné frameworky v teoretickej časti našej bakalárskej práce.

Pri programovaní aplikácie sme sa stretli aj s niekoľkými problémami. Jeden z nich sa objavil hneď na začiatku vývoja aplikácie. Bolo potrebné zistiť možnosti sťahovania obrázkov do aplikácie v reálnom čase, kedy sa daná aplikácia používa. Následne bolo potrebné toto sťahovanie vyriešiť tak, aby si užívateľ mohol vytvoriť a naplniť databázu objektov sám, bez potreby zasahovania do kódu aplikácie. S aplikáciou je možné pracovať v online režime. V mnohých dokumentáciách pracovali s objektami len lokálne, preto nám vyriešenie tohto problému trvalo najdlhšie. Nakoniec sa nám tento problém sťahovania z databázy a ladenie obrázkov v reálnom čase podarilo vyriešiť.

Práca na bakalárskej práci tohto zamerania bola pre nás veľmi prínosná. Naučili sme sa množstvo nových vedomostí a mali sme možnosť pracovať s doteraz pre nás novými technológiami. Aplikácia je možné použiť len na zariadenia podporujúce OpenGL ES 3.0 a vyššie. Tiež je potrebné použiť Android 8.0 a vyšší. V budúcnosti by sme chceli, do našej aplikácie doplniť možnosť nahrávania obrázkov a ich pozície na cloudové úložisko.

Rozšířená realita sa dnes stáva pomerne známym pojmom. Veľké spoločnosti si už adaptovali túto technológiu, aj vďaka tomu dnes existuje veľa kolektívnych informácií, na základe ktorých je možné vytvoriť napríklad mobilnú aplikáciu pre rozšírenú realitu. Rozšířená realita je nepochybne, jednou z technológií budúcnosti.

Literatúra

1. MADNIEKS, Zigurd; DORNIN, Laird; MEIKE, Blake; NAKAMURA, Masumi: *Programming Android*. O'Reilly Media, Inc., 2011. Java Programming for the New Generation of Mobile Devices. Dostupné z Google Books: 5BGBswAQSiEC.
2. MEIER: Reto. *Professional Android 2 Application Development*. John Wiley & Sons, 2010. Dostupné z Google Books: ZthJlG4o-2wC.
3. ARTH, Clemens; GRASSET, Raphael; GRUBER, Lukas; LANGLLOTZ, Tobias; MULLONI, Alessandro; WAGNER, Daniel: *The History of Mobile Augmented Reality*. Inst. for Computer Graphics a Vision Graz University of Technology, Austria, 2015. Developments in Mobile AR over the last almost 50 years. Dostupné z Research Gate: <https://www.researchgate.net/publication/275974448>.
4. *Building AR: 9 Best Frameworks for Developing Augmented Reality Apps* [online]. 2021 [cit. 2021-02-28]. Dostupné z: <https://saad-arshed.medium.com/best-frameworks-for-building-augmented-reality-apps-c5ec07575441>.
5. *UNITY: unity* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://unity.com/>.
6. *ARCORE: About ARCore* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://arvr.google.com/arcore/>.
7. *VUFORIA: Developer portal* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://developer.vuforia.com/>.
8. *WIKITUDE: Wikitude Augmented Reality SDK* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://www.wikitude.com/products/wikitude-sdk/>.
9. *EARYAR: Easy AR* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://www.easyar.com/>.
10. *KUDAN: Kudan AR SDK* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://www.xlsoft.com/en/products/kudan/ar-sdk.html>.
11. *ARTOOLKIT: Documentation Toolkit* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://www.xlsoft.com/en/products/kudan/ar-sdk.html>.
12. *ARKit: Introducing ARKit 4* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://developer.apple.com/augmented-reality/arkit/>.

13. *AR.JS: Ar.js* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://www.xlsoft.com/en/products/kudan/ar-sdk.html>.
14. *ARAPP: Best Tools for Building Augmented Reality Mobile Apps* [online]. 2021 [cit. 2021-01-26]. Dostupné z: <https://rubygarage.org/blog/best-tools-for-building-augmented-reality-mobile-apps>.
15. *UDEMY: unity* [online]. 2021 [cit. 2021-01-04]. Dostupné z: <https://www.udemy.com/course/arcore-and-sceneform-for-android-ar/>.
16. *UI: Čo je to UX a UI?* [Online]. 2021 [cit. 2021-01-04]. Dostupné z: <https://www.grappa.sk/blog/web-stranky/co-je-to-ux-a-ui-a-aky-je-medzi-nimi-rozdiel/>.
17. *ARFRAGMENT: Čo je to UX a UI?* [Online]. 2021 [cit. 2021-01-04]. Dostupné z: https://blog.griddynamics.com/latest-arcore-and-sceneform-features-take-creation-of-ar-apps-to-the-next-level/?fbclid=IwAR0RW4yPkej9m1lA1dCVyJQXDFV7zukeuh4mibeoA_APyHwJoGGdgTcVzZQ.
18. *FIREBASE: What is Firebase? The complete story, abridged.* [Online]. 2021 [cit. 2021-01-04]. Dostupné z: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>.